UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/711,317	09/10/2004	Arash Massoudi	16906.1.1	5316

22913 7590 10/05/2007

WORKMAN NYDEGGER
60 EAST SOUTH TEMPLE
1000 EAGLE GATE TOWER
SALT LAKE CITY, UT 84111

EXAMINER
TO, BAOQUOC N

ART UNIT	PAPER NUMBER
2162	

MAIL DATE	DELIVERY MODE
10/05/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/711,317	MASSOUDI	
	Examiner	Art Unit	
	Baoquoc N. To	2162	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *07/20/2007*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *13-37* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *13-37* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 0/13/2007 has been entered.

Claims 13-37 are pending in this application.

Response to Arguments

2. Applicant's arguments with respect to claims 13, 20 and 31 have been considered but are moot in view of the new ground(s) of rejection.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of

the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

3. Claims 13-37 are rejected under 35 U.S.C. 103(a) as being unpatentable over Vedula et al. (US. Patent No. 7,159,185 B1) in view of Sindhu et al. (US. Patent No. 6,917,620 B1).

As to claim 13, Vedula discloses in a computer network comprising two or more arbitrarily defined data structures, each of the data structures comprising one or more leaf data elements, a description of each data structure being known, a system for transferring data from a first in-memory data component corresponding to a first data structure to a second in-memory data component corresponding to a second data structure, the system comprising:

a) a mapping tool (mapping) (col. 9, lines 28-29) configured to:

allow a user to graphically define a relation and association between leaf data elements of a first data structure description and leaf data elements of a second data structure description (hierarchical tree structure) (col. 9, lines 276-28); and

generate one or more mapping descriptions of the relation and association between the leaf data elements of the first data structure description and the leaf data elements of the second data structure description as defined by the user (has created in mapping screen region 42) (col. 9, lines 28-29); and

b) a high-performance run-time engine (runtime engine) (col. 9, lines 32-33) configured to:

dynamically generate a first in-memory data component containing actual data associated with the leaf data elements of the first data structure description using a key-based look-up molding technique, the first in-memory data component comprising at least one lookup table (generated an source object, XML or other schema) (col. 9, lines 24-25);

dynamically generate a second in-memory data component configured to store actual data associated with the leaf data elements of the second data structure description using the key-based look-up molding technique, the second in-memory data component comprising at least one lookup table (translate source document into target or destination documents) (col. 9, lines 29-33); and

transfer data between the first in-memory data component and the second in-memory data component according to the one or more mapping descriptions (translation means to transfer one source to target) (col. 9, lines 29-33).

Vedula does not disclose traverse the one or more mapping descriptions and accessing the at least one lookup table of the first in-memory data

component to get actual data stored in the first in-memory data component using a key-based look-up technique. However, Sindhu discloses traverse the one or more mapping descriptions and accessing the at least one lookup table of the first in-memory data component to get actual data stored in the first in-memory data component using a key-based look-up technique (the controller include a key look-up engine and a route memory, the route memory storing a route table where the route table include a trie, the key look-up engine traversing the trie to determine a best match to the key...) (col. 35, lines 24-29). This suggests the concept traverse hierarchical structure by looking-up table in the memory. Therefore, it would have been obvious to one ordinary skill in the art at the time of the invention was made to modify Vedula to include traverse the hierarchical structure using looking-up table as disclosed Sindhu in order to allow the generation of the target objects.

As to claim 14, Vedula discloses the system as recited in claim 13, wherein the mapping tool is further configured to store the one or more mapping descriptions in a machine readable format (mapping) (col. 9, lines 29-30).

As to claim 15, Vedula discloses the system as recited in claim 14, wherein the machine readable format is XML (XML) (col. 9, line 24).

As to claim 16, Vedula discloses the system as recited in claim 13, wherein the one or more mapping descriptions is expressed using a unique path identifier with an absolute and relative path addressing scheme (mapping) (col. 9, lines 30-34).

As to claim 17, Vedula discloses the system as recited in claim 13, further comprising a cache configured to store the one or more mapping descriptions and configured to respond to multiple requests for accessing the one or more mapping descriptions (mapping) (col. 9, lines 30-34).

As to claim 18, Vedula discloses the system as recited in claim 13, wherein the mapping tool is further configured to allow a user to graphically define a one-to-one relation and association between the leaf data elements of the first data structure description and the leaf data elements of the second data structure description (mapping) (col. 9, lines 30-34).

As to claim 19, Vedula discloses the system as recited in claim 13, wherein the mapping tool is further configured to allow a user to graphically define a one-to-many relation and association between the leaf data elements of the first data structure description and the leaf data elements of the second data structure description (hierarchical mapping) (col. 9, lines 25-30).

As to claim 20, Vedula discloses a distributed computer system comprising one or more data structures, a method for storing and retrieving actual data of a data structure in order to use the actual data of the data structure to perform data transfer functions, the method comprising:

identifying a hierarchy of one or more data containers in a first data structure description, wherein a data container can be defined as either a singular data container or a plural data container, wherein at least one of the data elements in the one or more data containers is a leaf data element (object 26 and 30 in a hierarchical structure to allow the ease of mapping creation) (col. 9, lines 25-28);

determining whether all of the one or more data containers in the first data structure description are singular data containers, wherein, for each leaf data element, a key is generated containing a concatenation of all names of the data containers in a hierarchical path to the leaf data element, each data container name separated by a character that is not allowed as part of the data container name, concatenated with a name of the leaf data element, and storing the key in a lookup table of a single in-memory data component (an exemplary function object 100 is illustrated, having a script component 104 with computer executable instruction 106 for performing string concatenation function) (col. 9, lines 66-67 to col. 10, lines 1-2); or determining whether one or more data containers in the first data structure description is a plural data container, wherein upon identifying a plural data container, a component list is instantiated in a lookup table having a key that is generated containing a concatenation of names of all the data containers traversed either from a root node or from a previous plural data container to a hierarchical path to the identified plural data container, the component list comprising a plurality of data components.

Vedula does not explicitly disclose traversing the hierarchy of one or more data containers in the first data structure description to determine a unique key for each leaf data element of the first data structure description. However, Sindhu discloses traversing the hierarchy of one or more data containers in the first data structure description to determine a unique key for each leaf data element of the first data structure description (the controller include a key look-up engine and a route memory, the route memory storing a route table where the route table include a trie, the key look-up engine traversing the trie to determine a best match to the key...) (col. 35, lines 24-29). This suggests the concept traverse hierarchical structure by looking-up table in the memory. Therefore, it would have been obvious to one ordinary skill in the art at the time of the invention was made to modify Vedula to include traverse the hierarchical structure using looking-up table as disclosed Sindhu in order to allow the generation of the target objects.

As to claim 21, Vedula discloses the method as recited in claim 20, further comprising: upon identifying a plural data container, further comprising, for each leaf data element, generating a key containing a concatenation of names of all the data containers traversed from the previous plural data container to the hierarchical path of the leaf data element, each data container name separated by a character that is not allowed as part of the data container name, concatenated with a name of the leaf data element, and storing the key in one of the plurality of data components (concatenation function) (col. 10, lines 1-2).

As to claim 22, Vedula discloses the method as recited in claim 20, further comprising inserting a marker object as the value of the generated key (col. 10, lines 47-49).

As to claim 23, Vedula discloses the method as recited in claim 22, wherein the marker object is one of a string, a static object of a class (class) (col. 10, lines 25-26), or a unique integer value.

As to claim 24, Vedula discloses he method as recited in claim 20 excepting for further comprising: storing in cache one or more lookup tables containing the first data structure description; receiving a request for generating another lookup table for the first data structure description; and returning a copy of the cached one or more lookup tables for the first data structure description. Sindhu discloses storing in cache one or more lookup tables containing the first data structure description; receiving a request for generating another lookup table for the first data structure description; and returning a copy of the cached one or more lookup tables for the first data structure description (the controller include a key look-up engine and a route memory, the route memory storing a route table where the route table include a trie, the key look-up engine traversing the trie to determine a best match to the key…) (col. 35, lines 24-29). This suggests the concept generating lookup table. Therefore, it would have been obvious to one ordinary skill in the art at the time of the invention was made to modify Vedula to include generating looking-up table as disclosed Sindhu in order to allow the generation of the target objects.

As to claim 25, Vedula discloses the method as recited in claim 20, further comprising modifying actual data of at least one leaf data element of the first data structure description by including in a request a key generated for the leaf data element and sending the request to a lookup table of the leaf data element key to modify a data field corresponding to the leaf data element (col. 10, lines 47-54).

As to claim 26, Vedula discloses the method as recited in claim 20, further comprising using a mapping tool to define a relation and association between leaf data elements of the first data structure description and leaf data elements of a second data structure description (mapping) (col. 9, lines 28-29); and

automating transfer of the actual data associated with the leaf data elements of the first data structure description stored in the first in-memory data component to the second in-memory data component based on a mapping description of the relation and association between leaf data elements of the first data structure description and leaf data elements of the second data structure description, further comprising accessing one or more lookup tables of the first in-memory data component (translating to transfer data from source to target) (col. 9, lines 22-23).

As to claim 27, Vedula discloses the method as recited in claim 26, further comprising storing in memory the mapping description of the relation and association between leaf data elements of the first data structure description and leaf data elements

of the second data structure description (after translation data being stored) (col. 9, lines 33-36).

As to claim 28, Vedula discloses the method as recited in claim 26, wherein the mapping description of the relation and association between leaf data elements of the first data structure description and leaf data elements of the second data structure description includes identification of built-in-functions, further comprising transferring the actual data associated with the leaf data elements of the first data structure description stored in the first in-memory data component to the second in-memory data component using the built-in functions (col. 9, lines 31-33).

As to claim 29, Vedula discloses he method as recited in claim 26, wherein the mapping description of the relation and association between leaf data elements of the first data structure description and leaf data elements of the second data structure description includes leaf data elements from the first data structure description being flattened (col. 9, lines 23-25).

As to claim 30, Vedula discloses the method as recited in claim 20, further comprising enforcing a well-defined set of rules to restrict users as to data types that can be used to define the data elements of the first data structure description, the enforceable restrictions being one or more of whether the data type is singular or plural, a default value for the data type, whether the data type indicates that a corresponding

data value is required at runtime (run time) (col. 9, lines 30-31), a data range for the data type, allowed data values for the data type, a data format for the data type, or other enforceable restrictions for each data type.

Claim 31 is rejected as the same reason as to claim 1, Vedula also discloses a computer program product comprising computer usable medium having computer readable program coded stored thereon, the computer readable program code comprising computer readable instructions (col. 22, lines 39-41).

As to claim 32, Vedula teaches the method as recited in claim 31, wherein mapping a relation and association between leaf data elements of a first data structure description and leaf data elements of a second data structure description further comprises storing in memory the first data structure description in one or more lookup tables (generated an source object, XML or other schema) (col. 9, lines 24-25).

As to claim 33, Vedula teaches the method as recited in claim 31, wherein the mapping description is expressed in a machine readable format using a unique path identifier with an absolute and relative path addressing scheme (mapping) (col. 9, lines 29-30).

As to claim 34, Vedula teaches the method as recited in claim 31, wherein setting data values of any leaf data element of the first data structure description in a first in-memory data component using a key-based look-up molding technique further comprises storing a key corresponding to each leaf data element in one or more lookup tables (generated an source object, XML or other schema) (col. 9, lines 24-25).

As to claim 35, Vedula discloses the method as recited in claim 31, further comprising enforcing integrity of the data values while and after the data values are set (col. 9, lines 32-33).

As to claim 36, Vedula discloses the method as recited in claim 31, further comprising enforcing software interface integrity, data type validation and enforcing data type restrictions at runtime (col. 9, lines 32-33).

As to claim 37, Vedula discloses the method as recited in claim 31, wherein the first data structure and the second data structure are both software services, wherein mapping a relation and association between leaf data elements of a first data structure description and leaf data elements of a second data structure description comprises mapping the outputs of the first software service with the inputs of the second software service (translating from the source to the target) (col. 9, lines 32-33).

Conclusion

5. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Baoquoc N. To whose telephone number is at 571-272-4041, or unofficial fax number for the purpose of discussion (571) 273-4041 or via e-mail BaoquocN.To@uspto.gov. The examiner can normally be reached on Monday-Friday: 8:00 AM – 4:30 PM, EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John Breene can be reached at 571-272-4107.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703) 305-3900.

Any response to this action should be mailed to:

Commissioner of Patents and Trademarks

Washington, D.C. 20231.

The fax numbers for the organization where this application or proceeding is assigned are as follow:

(571) 273-8300 [Official Communication]

BQ To

September 30th, 2007